Paper submitted to

10^{th} Aachen Symposium on Signal Theory

Name:	Ian Herwono, Ingo Liebhardt
Affiliation:	Communication Networks Aachen University of Technology
Address:	Kopernikusstr. 16 D–52074 Aachen, Germany
Phone:	+49 241 80 27248
Fax.:	+49 241 80 22242
E-mail:	{ian,ilt}@comnets.rwth-aachen.de

Paper Title:	Performance Evaluation of the WAP Security Protocols
Symposium Topic:	Security and Privacy
Characterizing Keywords:	Security, Mobile Network, WAP, WTLS, Cryptography, Performance Evaluation
WWW:	http://www.comnets.rwth-aachen.de
Number of Pages:	7

Performance Evaluation of the WAP Security Protocols

Ian Herwono, Ingo Liebhardt

Communication Networks
Aachen University of Technology
Kopernikusstr. 16
D–52074 Aachen, Germany
Phone: +49 241 80 27248
E-mail: {ian,ilt}@comnets.rwth-aachen.de

Abstract

In this paper an overview of the *Wireless Application Protocol (WAP)* is given, whereby major work focuses on its security layer *WTLS*. With regard to the use of WTLS as a key technology for *Mobile Commerce*, the fulfillment of the corresponding cryptographic assurances is investigated. A simulation based performance evaluation of the employed security methods and algorithms is carried out in order to point out their advantages and disadvantages.

1 Introduction

Globally, 240 million[1] people are predicted to use their mobile phones for wireless data exchange by the end of 2004—up from 26 million in 1999. In order to prepare cellular phones and comparable devices such as pagers and personal digital assistants (PDAs), which are profoundly different from desktop computers for which the Internet was originally designed, for this kind of data transmission, the *Wireless Application Protocol (WAP)* has been introduced. By adapting the existing network technology to the new requirements, WAP specifies an application framework and network protocols for wireless devices. Being positioned at the convergence of the two rapidly evolving technologies wireless data and Internet, WAP has to cope with a more constrained computing environment compared to desktop computers. Because of fundamental limitations of power and form-factor, mass-market hand-held devices tend to have:

- Less powerful CPUs and less memory (ROM and RAM),
- Smaller displays and input devices.

[1] according to Allied Business Intelligence

Similarly, wireless data networks present a more constrained communication environment compared to wired networks and tend to have:

- Less bandwidth and more latency,
- Less connection stability, less predictable availability.

The above mentioned limitations are to be taken into account especially when providing a secure transport service for the upper layers of the WAP protocol stack, e. g. , for enabling the user to perform *Mobile Commerce* transactions, as security-related operations usually consume more resources.

While the process of specification, which is supervised by the WAP-Forum, is presently going on, this work gives an overview of the protocol stack as specified in the current version (1. 2. 1) of WAP. After introducing the five protocol layers the work focuses on *security aspects* and introduces cryptographic assurances needed to be fulfilled by *any* complete security system. Following this, we show in how far these assurances are met by WTLS—WAP's security layer—and finally present some simulation results to depict its expected performance.

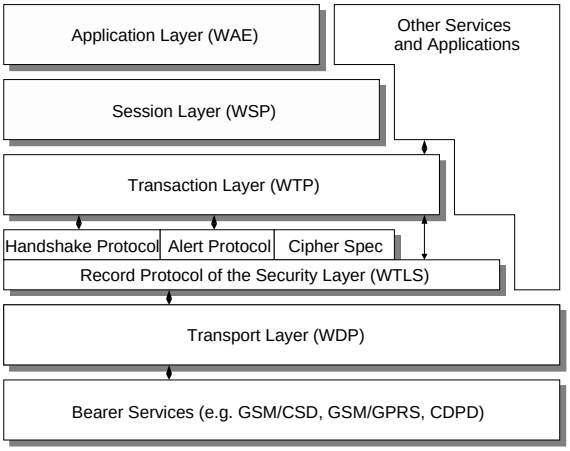

Figure 1: The WAP protocol stack

2 WAP – An Overview

The topmost layer called Wireless Application Environment (WAE) includes an inter-operable micro-browser with an own markup- and script-language. The Wireless Session Protocol (WSP) offers services suited for browsing applications including HTTP/1.1 functionality. The Wireless Transaction Protocol (WTP) provides as a lightweight transaction-oriented protocol that is suitable for performing unreliable or reliable transport of data. Wireless Transport Layer Security (WTLS) is a security protocol based upon the industry-standard Transport Layer Security (TLS) protocol. The Wireless Datagram Protocol (WDP) operates above the data capable bearer services and offers a consistent service to the upper layer protocols of WAP. Amongst others, GSM/CSD, GSM/GPRS, CDPD and even SMS can be used as bearer services. Fig. 1 gives an overview of the entire protocol stack [1].

Additionally a Wireless Identity Module (WIM), which keeps the user's private key under closure and performs the computations needed for public-key cryptography, is defined. It may be implemented as a separate WIM card, a combined SIM/WIM card, another fixed or removable device carrying WIM-functionality, or a software solution.

3 Cryptographic Assurances

Any complete electronic security system used for data-transmission has to fulfil certain cryptographic assurances [2]. The first, *confidentiality*, is the assurance that only owners of a shared secret key can decrypt the data that has been encrypted with the identical secret key with a reasonable effort. It is already this first assurance that causes a problem: how can two parties start communication and establish a shared secret without allowing other persons to ascertain the secret key. In order to solve this problem, public-key cryptography—also called asymmetric cryptography because the keys used for enciphering and deciphering are different from each other—has been introduced. The second assurance is called *integrity* or *message authentication*. It assures the receiver that the data received is exactly the data originally transmitted by the sender and that no intentional changes have been performed during the transmission. Another cryptographic assurance is *authentication*, that enables both parties to identify the partner of communication securely and hence prevents masquerading. *Nonrepudiation*, the last cryptographic assurance, takes care that the sender cannot deny a message sent by herself. Exactly like a signature on paper, a so-called digital signature proves that a message can only origin from this particular sender, which is vital for any kind of electronic or mobile commerce.

4 Wireless Transport Layer Security

The WTLS, WAP's security layer, offers numerous cryptographic algorithms to meet the assurances specified above. To provide confidentiality several symmetric algorithms like DES, 3DES, RC5, or IDEA are imparted. For ensuring message authentication, a keyed HMAC hash is used in combination with MD5 or SHA-1. RSA and ECDH are suggested for the key exchange process. Because of this versatility a handshake procedure, during which the communicating parties agree upon the used algorithms and a shared secret, is needed.

WTLS is a layered protocol consisting of the record layer and the handshake protocol, which again includes the two sub-protocols 'change cipher spec protocol' and 'alert protocol'. The change cipher spec protocol is used to indicate a change in encryption parameters to the opposite party's record layer and the alert protocol is used to report and handle error conditions. The other two sublayers are described in the following.

4.1 Handshake Protocol

A *full handshake* starts with the client sending a *client-hello* message in which it announces its supported algorithms and parameters in the order of the client's preference. For simplicity, numbers are assigned to these algorithms and they are grouped into two suites the first of which—the so-called key-exchange-suite—contains the algorithms needed for key exchange and authentication. The second one—the cipher-suite—contains the algorithms needed for en- and deciphering as well as for calculating the keyed message authentication code (MAC).

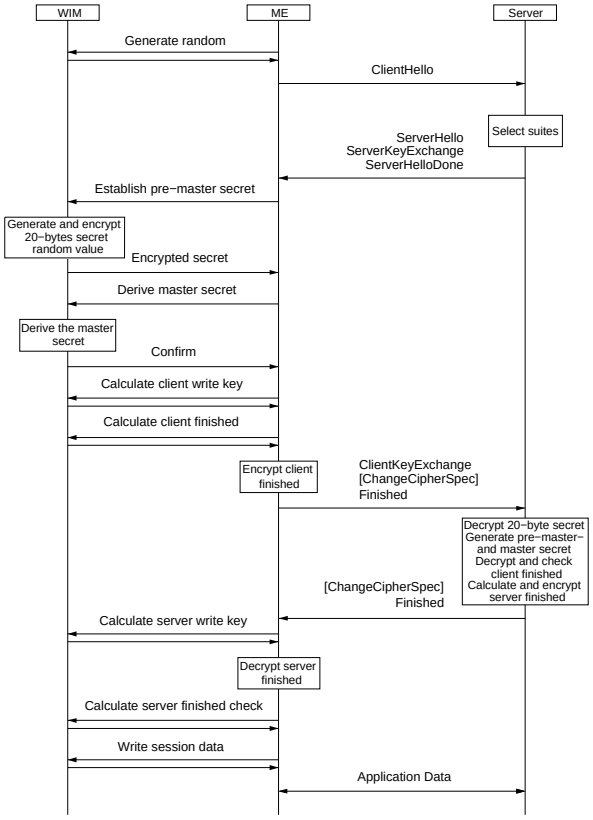

Figure 2: Message flow for an anonymous RSA Handshake

After this the server answers with a *server-hello* message in which it transmits the algorithms and cryptographic parameters it has chosen for this connection, appends its public key or certificate, depending on the chosen key exchange suite, concatenates these messages together with a *server-finished* message into one transport SDU[2] and transmits it to the client.

In case the chosen key-exchange-suite implies *RSA*, the client generates a 20-byte secret value, encrypts it under the server's public key and sends it to the server. Once encrypted with the public key, the secret value can only be decrypted with the server's private key and can therefore be used as the *pre_master_secret*, which is only known to the two communicating parties. Fig. 2 shows the message flow for an *anonymous* RSA-type handshake, i. e. , without exchange of certificates.

In case the key-exchange-suite implies *ECDH* each party sends its public key (i. e. a point of a commonly known elliptic curve E) to the respective opposite party. Then a modified Diffie-Hellman key-agreement scheme adapted to the needs of elliptic curve cryptography is used to agree upon the same secret value. The key-agreement scheme requires a common point P of an elliptic curve E known to all parties. The public key sent by the server is a new point Q_A generated by the server by multiplying a random value a by the shared point P: $Q_A = aP$. The client does the same with its random value b and sends this point $Q_B = bP$ to the server. Now the server multiplies this received value Q_B by its random value a and the client multiplies its received value Q_A by its random value b. Based on the elliptic curve theory it follows that

$$k = a(bP) = b(aP) = k'.$$

As eavesdroppers cannot easily derive k from Q_A and Q_B[3], the agreed upon value k can be used as the *pre_master_secret*.

Once the *pre_master_secret* has been established in both parties, a pseudo-random function (PRF) is used to generate the *master_secret* and the secret keys used for the symmetric cipher suite are again generated out of the *master_secret* with the help of the irreversible PRF.

An *abbreviated handshake* can be applied if both communicating parties already had completed a full handshake earlier. In this case, client and server retrieve the old *pre_master_secret* out of the WIM and a cache respectively and the application data can be transferred directly after the *ChangeCipherSpec* and *Finished* messages have been exchanged.

After the handshake has been completed successfully, the transparent transmission of confidential user data can take place. Confidentiality is granted by enciphering the data for transmission, and integrity is granted by applying a keyed HMAC-hash function.

4.2 Record Protocol

The record protocol is responsible for the computation of the keyed MAC, the padding of the data obtained by appending the MAC to the content, and the encryption of the resulting structure (Fig. 3). Received data is of course decrypted, verified and then delivered to the appropriate higher level client, i. e. , the handshake protocol, the change cipher spec protocol, the alert protocol, or the application data protocol (Fig. 1). All en- and decryption is performed by ciphers operating in Cipher Block Chaining (CBC) mode.

The key material needed for MAC-calculation, encryption, and provision of the initialisation vector (IV) is periodically refreshed by means of the PRF, which is irreversible in order to ensure that the long-lasting *master_secret* cannot be found even if the temporary encryption key has been compromised.

In future versions of WTLS the use of stream ciphers as well as compression/decompression of user data are planned [4].

[2]Service Data Unit

[3]Deriving the agreed upon value from a public key is known as the Elliptic Curve Discrete Logarithm Problem (ECDLP)[3].

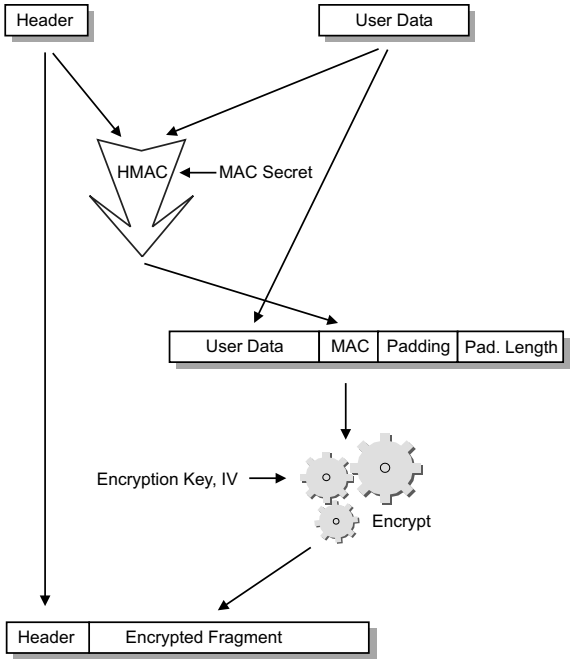

Figure 3: Packet data flow in the WTLS Record Protocol

By protecting the payload with the mechanisms specified above, WTLS grants for authentication, confidentiality, and integrity. Note that WTLS is unable to assure nonrepudiation.

5 Results

A simulation tool for the performance evaluation of WAP security protocols has been developed at the Chair of Communication Networks at the Aachen University of Technology. It is a prototypical, standard-conformant implementation of WTLS, including WDP and IP as the underlying layers. The simulator is formally specified in SDL[4] and coded using C/C++. Most of the implementations concerning cryptographic computations originate from the free C++ class library Crypto++ 4.1[5]. All following measurements are the results of tests carried out on a SUN Enterprise server equipped with 1664 Mbyte RAM and processors of 400 MHz clock frequency, whereby only CPU times have been taken into account.

5.1 WTLS Handshake Protocol

The overall durations of several handshake procedures have been measured while varying the effective mean throughput of the underlying bearer[6], which is determined by numerous factors such as available radio resources, network latency and channel quality. These measurements allow the comparison of the two competing public-key cryptosystems RSA and ECDH. Furthermore handshakes employing different lengths of public keys have been examined.

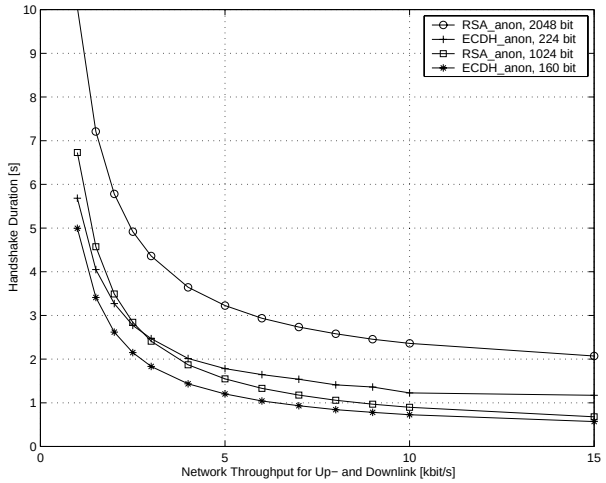

Figure 4: Duration of examined WTLS Handshakes

In Fig. 4 the overall durations of RSA handshakes with key lengths of 1024 and 2048 bits as well as the durations of ECDH handshakes with key lengths of 160 and 224 bits are depicted. Note that the security levels provided by 1024 bit and 2048 bit RSA keys are roughly comparable with those ones provided by 160 bit and 224 bit ECDH keys respectively. Given that the network throughput is low ECDH-type handshakes are more advantageous. The higher the security requirements are, the more this effect will appear. The impact of the various cryptographic methods becomes negligible as the network throughput decreases, and the amount of data[7] transferred across the air interface during a single handshake (Table 1) gets more important.

Table 1: Size of message groups exchanged during a handshake

	RSA (1024 bit)	**ECDH (160 bit)**	**RSA (2048 bit)**	**ECDH (224 bit)**
1st msg.	263 bytes	263 bytes	263 bytes	263 bytes
2nd msg.	210 bytes	98 bytes	338 bytes	106 bytes
3rd msg.	221 bytes	115 bytes	349 bytes	123 bytes
4th msg.	116 bytes	116 bytes	116 bytes	116 bytes

[4] Specification and Description Language

[5] Please refer to `http://www.eskimo.com/~weidai/cryptlib.html` for further information.

[6] assuming a symmetrical channel, i.e., rates in uplink and downlink are identical

[7] includes WDP- and IP-overhead

5.2 WTLS Record Protocol

The throughput of the WTLS layer has been measured on the supposition that it is only limited by the processing power of the client and the server respectively. In order to allow comparison, the throughput of several symmetric-key ciphers in combination with the two variations of the HMAC-hash functions, i. e. , SHA-1 and MD5, has been investigated. The measurements included all operations necessary to fulfil the record sub-layer's requirements stated in [4]. Besides en- and decryption, this also includes the application, recalculation and comparison of the MAC, computation of the record IV, and some relocations of data in memory.

For ensuring realistic conditions, all tests have been performed by operating the ciphers on real user data rather than operating them on one block of data repeatedly. Hence most of the data had to be retrieved out of the server's RAM instead of its cache.

Note that some of the implemented symmetric ciphers, e. g. , AES, are not included in the WTLS standard yet. DES, 3DES, RC5, and IDEA have been tested with a blocksize of 64 bit whereas a blocksize of 128 bit has been chosen for AES (Rijndael). Encryption and decryption performance of each algorithm have been evaluated with several packet[8] sizes ranging from 256 bytes up to 8192 bytes and the resulting throughputs have been averaged. Note that only user data has been taken into account for calculating the throughput whilst the encrypted data additionally includes the MAC and padding.

Table 2: Throughput of the WTLS-layer in Mbit/s

keyed MAC	DES	3DES	RC5	IDEA	AES
MD5	5.78	2.38	8.85	6.14	14.93
SHA-1	5.14	2.26	7.43	5.38	10.96

Table 2 shows that AES (Rijndael) in combination with MD5 provides the highest performance while ensuring a very good grade of security as well. As expected, 3DES shows the lowest performance when compared to newer ciphers providing the same grade of security. However, the performance measurement is influenced by the effectiveness of the software implementation.

6 Conclusions

Our performance evaluation has shown that once the handshake procedure has been completed, the impact of the WTLS record protocol on the timing behaviour of the WAP protocol stack is negligible, as long as only clients, e. g. , mobile phones, are concerned. Although a powerful workstation has been employed, comparable en- and decryption rates may be achieved in hardware implementations that could be integrated in future mobile appliances [5].

Contrary to the throughput of the security layer, attention has to be paid to the duration of a handshake. With respect to the use of WTLS for securing M-Commerce transactions, performing a full handshake prior to each transaction may cause a significant increase in the overall transaction duration. Replacing subsequent full handshakes with abbreviated ones will considerably accelerate the completion. However this measure might weaken the security level since the longer lifetime of the master secret makes it more vulnerable. Therefore a compromise between the provided security and the transaction performance has to be found so as to foster user acceptance.

[8] WTLS SDU

References

[1] WAP Forum, *Wireless Application Protocol Architecture Specification*, (URL: http: //www.wapforum.org/), April 1998.

[2] D. Baker and H. X. Mel, *Cryptography Decrypted*. Addison-Wesley, 2000.

[3] Certicom Corp., *Remarks on the Security of the Elliptic Curve Cryptosystem*, (URL: http: //www.certicom.org/), July 2000.

[4] WAP Forum, *Wireless Transport Layer Security Protocol*, (URL: http: //www.wapforum.org/), February 2000.

[5] Infineon Technologies AG, *Security & Chip Card ICs, SLE 66CX160S*, (URL: http: //www.infineon.com/), September 2000.

[6] I. Liebhardt, *Development of a Simulation Platform for the Performance Evaluation of WAP-based Mobile Commerce Services*. Master's Thesis, RWTH Aachen, Lehrstuhl für Kommunikationsnetze, May 2001.